

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-333838

(43)Date of publication of application : 18.12.1998

(51)Int.Cl.

G06F 3/06

(21)Application number : 09-146652

(71)Applicant : HITACHI LTD

(22)Date of filing : 04.06.1997

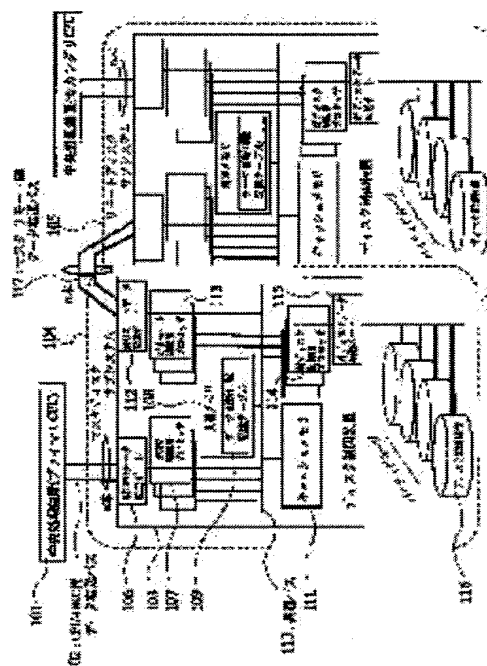
(72)Inventor : AZUMI YOSHIHIRO
IZUMI HIROYUKI
NAKANISHI HIROAKI

(54) DATA MULTIPLEXING STORAGE SUB-SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To enhance the processing performance of each storage sub-system, by suppressing the increase of a load accompanying data copy for data multiplexing between plural storage sub-systems.

SOLUTION: In this storage sub-system, a master disk sub-system 104, which has a constitution connecting disk storage devices 116 under the command of a disk controller 103, is connected through an inter-master and remote data transfer path 117 with a remote disk sub-system 105 in the same constitution, and write data outputted from a central processing unit 101 through a data transfer path 102 to the master disk sub-system 104 are copied to the remote disk sub-system 105. In this case, this storage sub-system is provided with a data update under of time storage table 109 which manages a data update history for each data management unit such as a track, cylinder, volume, and file, and the continuity of the data update is judged for each data management unit, and an executing equipment is controlled for summarizing the data copying operation so that a load resulted from the execution of the data copy can be reduced.



LEGAL STATUS

[Date of request for examination] 13.09.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3400297

[Date of registration] 21.02.2003

[Number of appeal against examiner's decision]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-333838

(43) 公開日 平成10年(1998)12月18日

(51) Int.Cl.⁶

G 0 6 F 3/06

識別記号

3 0 4

F I

G 0 6 F 3/06

3 0 4 E

審査請求 未請求 請求項の数 3 O L (全 15 頁)

(21) 出願番号 特願平9-146652

(22) 出願日 平成9年(1997)6月4日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 安積 義弘

神奈川県小田原市国府津2880番地 株式会

社日立製作所ストレージシステム事業部内

(72) 発明者 泉 洋行

神奈川県小田原市国府津2880番地 株式会

社日立製作所ストレージシステム事業部内

(72) 発明者 中西 弘晃

神奈川県小田原市国府津2880番地 株式会

社日立製作所ストレージシステム事業部内

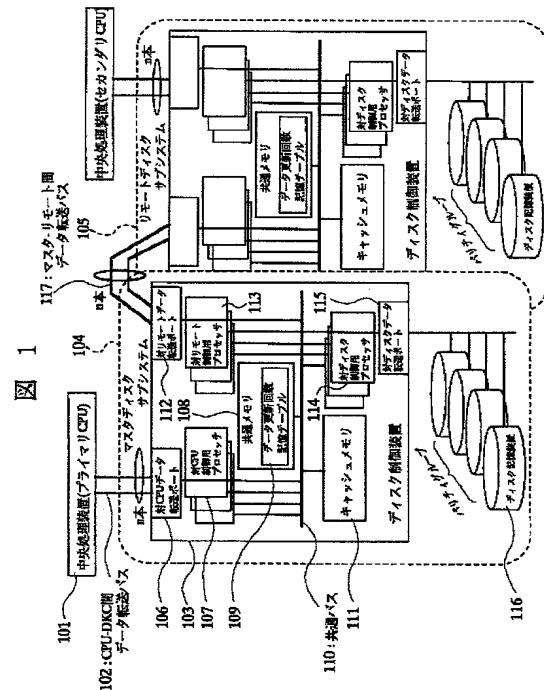
(74) 代理人 弁理士 筒井 大和

(54) 【発明の名称】 データ多重化記憶サブシステム

(57) 【要約】

【課題】 複数の記憶サブシステム間でのデータ多重化のためのデータ複写に伴う負荷の増大を抑制して各記憶サブシステムの処理性能を向上させる。

【解決手段】 ディスク制御装置103の配下にディスク記憶装置116を接続した構成のマスタディスクサブシステム104と、これと同一構成のリモートディスクサブシステム105とをマスタ-リモート間データ転送パス117にて接続し、データ転送パス102を介して中央処理装置101からマスタディスクサブシステム104に出力されるライトデータをリモートディスクサブシステム105に複写する構成において、トラック、シリンダ、ボリューム、ファイル等のデータ管理単位毎のデータ更新履歴を管理するデータ更新回数記憶テーブル109を設け、データ管理単位ごとにデータ更新の継続性を判断してデータ複写操作が集約されるように実行契機を制御して、データ複写の実行に起因する負荷を軽減する。



【 特許請求の範囲】

【 請求項1 】 上位装置と第1 のデータ転送経路を介して接続された記憶制御装置および当該記憶制御装置の配下の記憶装置を含む第1 の記憶サブシステムと、各々が、記憶制御装置および当該記憶制御装置の配下の記憶装置を含み、第2 のデータ転送経路を介して前記第1 の記憶サブシステムに接続された少なくとも一つの第2 の記憶サブシステムと、からなり、前記第1 のデータ転送経路を介して前記上位装置から受領したライト データを前記第1 の記憶サブシステム内の前記記憶装置に格納するとともに、任意の契機にて前記第2 のデータ転送経路を介して前記第2 の記憶サブシステム内の前記記憶装置に複写することによって、前記第1 および第2 の記憶サブシステムにて前記ライト データを多重に保持するデータ多重化記憶サブシステムであって、前記上位装置から到来する前記ライト データの前記第1 の記憶サブシステムの前記記憶装置における所望のデータ管理単位に対する 書き込み履歴を記憶する制御情報記憶手段と、前記制御情報記憶手段の前記書き込み履歴を参照し、前記上位装置から前記データ管理単位に対する単位時間当たりのデータ書き込み要求の偏りを観測して、前記第2 の記憶サブシステムに対する前記ライト データの前記複写の実行方法および実行契機の少なくとも一方を制御する制御論理と、を備えたことを特徴とするデータ多重化記憶サブシステム。

【 請求項2 】 請求項1 記載のデータ多重化記憶サブシステムにおいて、前記書き込み履歴を記憶する前記制御情報記憶手段として複数の記憶テーブルを備え、所望の周期毎に前記記憶テーブルを切替えることで、時間の経過に対する前記上位装置からのデータ書き込み要求の変化を観測し、観測された前記データ書き込み要求の変化に応じて、前記上位装置からの前記データ書き込み要求の継続性を推測し、前記第2 の記憶サブシステムに対する前記ライト データの前記複写の実行契機を制御する第1 の制御動作、前記第1 の記憶サブシステムのデータを一括して前記第2 の記憶サブシステムに複写する際に、前記制御情報記憶手段に格納されている前記書き込み履歴を参照することによって、複写対象範囲に対する前記上位装置からのデータ書き込み傾向を観測し、前記上位装置からのデータ書き込み要求が継続または集中している未複写の前記データ管理単位に対する前記複写の実行契機を遅らせる第2 の制御動作、の少なくとも一方の制御動作を行うことを特徴とするデータ多重化記憶サブシステム。

【 請求項3 】 請求項1 記載のデータ多重化記憶サブシステムにおいて、前記第1 および第2 の記憶サブシステムの各々の前記記憶装置は、グループを構成する複数の

データ単位および当該データ単位から生成された冗長データを複数の記憶媒体に分散して格納する論理的または物理的な冗長記憶構成を備え、

前記書き込み履歴を記憶する前記制御情報記憶手段を参照して個々の前記グループ内の前記データ単位に対するデータ書き込み要求の発生状態を観測し、当該グループ内の全ての前記データ単位を一括して前記第2 の記憶サブシステムに転送する第1 の複写操作、および前記グループ内の複数の前記データ単位のうち、前記上位装置からのデータ書き込み要求によって更新された前記データ単位のみを選択的に前記第2 の記憶サブシステムに転送する第2 の複写操作、の選択および実行契機を制御する第3 の制御動作、

前記第3 の制御動作において、前記第1 および第2 の前記記憶サブシステム間におけるデータ転送所要時間を含めた前記複写操作の所要時間を観測し、前記第1 および第2 の複写操作の選択および実行契機の制御を行う第4 の制御動作、

の少なくとも一方の制御動作を行うことを特徴とするデータ多重化記憶サブシステム。

【 発明の詳細な説明】

【 0001 】

【 発明の属する技術分野】 本発明は、データ多重化記憶技術に関し、特に、遠隔地等に独立に分散して配置された複数の記憶サブシステムにて同一データを多重に分散して保持することでデータ保障を実現する情報処理システム等に適用して有効な技術に関する。

【 0002 】

【 従来の技術】 中央処理装置とディスク記憶装置に代表される周辺記憶装置(主にディスク記憶装置とディスク制御装置とからなるディスクサブシステム)とからなる情報処理システムでは、情報量の膨大化とともに取り扱うデータの記憶に対する信頼性への要求が強まる中で、従来よりディスク装置などの記憶媒体や記憶装置の物理的な障害に対する信頼性向上策として、複数の記憶媒体にデータを二重に保持することによって、障害に伴うデータ消失に対しバックアップデータからの回復を図るデータ二重化記憶サブシステムが実用化されている。また、データを複数のディスク装置に分割して配置し、更に幾つかのデータを一単位としてパリティデータに代表される冗長データを作成・記憶することによって、あるデータの媒体障害やディスク装置の障害時に、冗長データと当該一単位内の他データとからデータ回復を行なうRAID記憶装置も実用化されている。

【 0003 】ところが銀行等のオンラインシステムに代表されるように、広域に渡って情報処理システムが機能し、多くの情報処理システムが連動しているようなシステムにおいては、これらのデータ信頼性向上技術は、一つの記憶サブシステム内でデータを多重に保持したり冗長化を図るものであり、その記憶サブシステム全体の障

10

20

30

40

50

害や、中央処理装置を含む情報処理システム全体がたとえば建物全体の停電・火災等によって動作しなくなった場合、その被害が広域のシステム全体に影響を及ぼすばかりでなく、データ消失に伴う被害度は甚大なものになってしまう。この様な懸念に対し、遠隔地においてデータを二重に保持するデータ二重化管理システムが実用化されている。しかしながらこの遠隔データ二重化においては、遠隔地に設置された情報処理システム間のデータの通信を中央処理装置間の通信機能によって処理しているため、データ処理や演算等を行なう中央処理装置の負荷が大きく、この中央処理装置の負荷を軽減することが、遠隔データ二重化システムの課題とされている。

【0004】この様な課題に対し、ディスク制御装置に制御装置間で通信およびデータ転送を行なう機能を設け、遠隔地にあるそれぞれの情報処理システムの制御装置同士を、通信・データ転送パスで接続することにより、データ二重化に掛かる負荷を記憶制御装置に負担させることで中央処理装置の負荷を軽減するシステムも実用化されている。この遠隔データ二重化記憶サブシステムでは、主業務を行なう情報処理システムをプライマリシステムとし、それぞれ第1の中央処理装置、第1のディスクサブシステムおよび第1のディスク制御装置とする。また、バックアップ側の情報処理システムをセカンダリシステムとし、それぞれ第2の中央処理装置、第2のディスクサブシステムおよび第2のディスク制御装置とする。第1、第2のそれぞれのディスク制御装置は不揮発化機構を備えた大容量のキャッシュメモリ（ディスクキャッシュ）を備えている場合が一般的である。第1と第2のディスク制御装置間を1本ないしは複数本のデータ転送パスで接続し、データの一単位毎（たとえばボリューム毎）に正・副のペアボリュームの関係を定義する。正側のデータをマスタデータ（マスタボリューム）呼び、副側データをリモートデータ（リモートボリューム）と呼ぶ。プライマリシステムでのディスクサブシステムへのWRT_I/Oにおいては、第1の中央処理装置から第1のディスクサブシステムへの書き込みデータを、自配下のディスク記憶装置に書き込むだけでなく、第2のディスク制御装置のホストとして第2のディスクサブシステムにデータ書き込みI/Oを発行し、データの二重化を図る。この様にしてデータファイルの二重化の運用を行なっている最中に、プライマリシステム側で障害が発生し、業務の継続が不可能になった場合には、即座にセカンダリシステムに業務を切替え、二重化されている第2のディスクサブシステムのデータを元に業務を継続する。

【0005】なお、データの二重化技術としては、たとえば、米国特許第5,155,845号に開示される技術が知られている。この技術では、分散して配置された複数の制御ユニットと、各制御ユニットの配下に等価な構成で接続された記憶手段とを設け、ひとつの制御ユニ

ットがレコードの書き込み要求を受けると他の各制御ユニット配下の対応するすべてのボリュームに当該レコードのコピーが書き込まれるようにしたものである。

【0006】前述の第1のディスク制御装置からのWRT_I/Oによるデータ二重化においては、データ二重化の契機に関し、主に以下の二通りの方式がある。

【0007】(1) 同期方式

プライマリシステムの第1の中央処理装置から第1のディスクサブシステムへのWRT_I/Oに同期して、第2のディスクサブシステムに同一データのWRT_I/Oを発行することによって、プライマリ側のデータとセカンダリ側のデータが常に同期しているように制御する方式。第1のディスク制御装置は、第1の中央処理装置からのWRT_I/O時に、自制御装置内のキャッシュメモリにWRTデータを書き込んだ時点で、データ転送の完了報告を行ない、その後第2のディスクサブシステムに同一のWRT_I/Oを発行し自キャッシュメモリ上のデータを第2のディスク制御装置に転送することによって、二重化のためのWRT_I/O処理を行なう。第2のディスクサブシステムへのWRT_I/Oが完了した時点で、第1のディスク制御装置は第1の中央処理装置にI/O完了報告を行なう。即ち、第1の中央処理装置がWRT_I/Oの完了報告を受領した時点で、第2のディスクサブシステムへのデータ複写は完了しているため、プライマリ側のデータとセカンダリ側のデータの同期性は保たれる。

【0008】(2) 非同期方式

第1のディスクサブシステムのデータ更新に対して、第2のディスクサブシステムへのデータの更新を非同期に行なう方式。第1のディスク制御装置は第1の中央処理装置からのWRT_I/O時に、第1のディスクサブシステムにデータを書き込んだだけでI/O完了報告を行なう。第1のディスクサブシステムへのデータの書き込みに関しては、ディスク記憶装置の記憶媒体へのデータ書き込みが終わってからI/O完了報告としても良いし、ディスク制御装置内のキャッシュメモリにデータを格納しただけでI/O完了報告を行なっても良い。第1のディスクサブシステムに書き込まれたが第2のディスクサブシステムに対してはデータの反映を行っていないデータは、第1のディスク制御装置にて未反映データとして管理される。第1のディスク制御装置は、一定周期や中央処理装置からのデータ反映要求、もしくは未反映データの残留量に応じて、中央処理装置からのWRT_I/Oとは非同期に、第2のディスクサブシステムに対してWRT_I/Oを起動し、未反映データの書き込みを行なう。

【0009】また、既に第1のディスクサブシステム上に存在するデータボリュームを新たに遠隔二重化ボリュームとして定義し二重化ペアを新規に作成する場合（これを初期コピーと呼ぶ）には、第1のディスク制御装置

10

20

30

40

50

は、第1のディスクサブシステムの当該ボリュームのデータを順次にディスク記憶装置からキャッシュメモリに読み出し、第1のディスク制御装置から第2のディスクサブシステムに書き込みI/Oを発行することによって、ボリュームデータの複写を行なう。この時のデータ複写の一単位はデータ格納単位の一単位(トラック)毎であっても良いし、複数のデータ単位(たとえば、シリンダ)毎であっても構わない。

【0010】更に、第1のディスクサブシステムは、初期コピー処理のI/Oを実行しながら、同時に第1の中央処理装置からの更新I/Oを受けることも可能である。初期コピー実行中のボリューム上のデータに対する更新においては、第1のディスク制御装置は、その更新範囲が、初期コピー処理が実施済み(第2のディスクサブシステムへの複写が完了済み)の領域に対する更新の場合には、同期または非同期の方式において第2のディスクサブシステムへの更新データの反映を行なう。また、更新範囲が初期コピー未実施の領域に対する更新の場合には、いずれ初期コピーのための二重化WRT_I/Oによって第2のディスクサブシステムへのデータ複写が行われるので、第1のディスクサブシステムへのデータ更新のみであっても構わない。

【0011】ところで、第1および第2のディスクサブシステムは、以下に述べるようなRAID-5のデータ格納方式であっても良い。本技術は、D. A. Patterson, et, al. "Introduction to Redundant Arrays of Inexpensive Disks (RAID)", spring COMP CON'89, pp. 112-117, Feb. 1989の論文にて述べられている技術である。RAID-5とは、ディスクサブシステムをn+m個のディスク記憶装置を一つのデータ格納単位とし、データのある一単位(たとえば、ディスク媒体上の1トラック)毎に、n個のディスク記憶装置に分割して格納する。さらにn個のデータ単位を1グループとしてパリティデータと呼ばれる冗長データを作成する。冗長データ数はその冗長度に応じて定まり、冗長度がmの場合はm個の冗長データを作成する。冗長データそのものも当該冗長データを構成するデータグループの格納ディスク装置とはまた異なるディスク装置に格納する。このn個のデータ単位とそのm個の冗長データから構成されるデータ群を冗長化グループと呼ぶ。このことにより、一つのディスク記憶装置が障害により読み出し不能に陥ったとしても、当該冗長化グループの他のn-1個のデータとm個の冗長データからデータの再生が可能であり、また同様に障害によって書き込み不良に陥った場合でもm個の冗長データを更新しておくことで論理的にデータの格納がなされる。このようにしてディスク装置やディスク媒体の障害に対しデータの信頼性を高めている。さて、RAID-5のデータ記憶方式においては、データ

の更新に際し、主に以下の2通りの冗長データ作成方法がある。

【0012】(1) 全ストライプライト方式

冗長化グループを構成するデータ単位グループをストライプ列と定義し、これらの全データ単位から冗長データを新たに作り出す方式。

【0013】(2) リードモディファイライト方式

冗長化グループを構成するデータ単位のある一単位が更新された場合に、更新データ単位の旧データと更新データと旧冗長データとを演算し、新冗長データを作成する方式。中央処理装置からのある一単位データの更新時に、ディスク制御装置はキャッシュメモリ上に旧データと新データを保持し、また当該冗長化グループの冗長データがキャッシュメモリ上に存在しない場合には、冗長化データをディスク記憶装置からキャッシュメモリ上に読み出し、新冗長データを作成する。この様にデータ単位の更新に対し余分に冗長データのディスク装置からの読み出し・書き込みが発生することをライトペナルティと呼ぶ。

【0014】

【発明が解決しようとする課題】上述の遠隔データ二重化においては、同期方式の二重化を採用した場合、第1の中央処理装置からのWRT_I/O時の応答時間は、I/O完了報告前に第2のディスクサブシステムへのデータ書き込みI/Oを行なうために、約二倍の処理時間が必要となる。また、非同期方式の二重化を採用した場合においても、中央処理装置からのWRT_I/Oの応答時間そのものは維持されるものの、ディスク制御装置のスループットは二重化のためのI/O処理の負荷により劣化は免れ得ない。このため、遠隔二重化のシステムにおいては、ディスク制御装置のデータ二重化処理の効率向上が性能上の最大の技術的課題となる。

【0015】中央処理装置からのボリュームデータの更新処理は、その形態によってはある特定の領域にアクセスが集中し、たとえばトラックやシリンダ等のデータ単位に対して繰り返し更新を行なう場合もある。このようなボリュームデータの更新形態においては、同期式の二重化方式とした場合、中央処理装置からのデータ更新回数と同一の回数だけ第2のディスクサブシステムへのWRT_I/Oが必要となる。一方、非同期方式の二重化方式とした場合、ある期間第1のディスク制御装置に未反映データを滞留させることによって、二重化を図る際の最新データのみをまとめて反映させれば良いため、第2のディスクサブシステムへのWRT_I/Oの発行回数を、第1の中央処理装置からの第1のディスクサブシステムへのWRT_I/O回数より削減することが可能である。非同期方式の二重化方式においては、いかに効率よく二重化データをまとめるかが性能向上の最大のポイントとなる。必要以上に第1のディスクサブシステム内に未反映データを滞留させることは、逆にキャッシュメ

モリ の利用効率を下げ、性能劣化の要因となり得るからである。

【0016】また、RAID-5 の記憶方式の場合、前述の全ストライプ方式の冗長データ作成方式とリードモディファイライト方式の冗長データ作成方式とでは、明らかに冗長データの作成効率に差が生じる。即ち、全ストライプ方式の冗長データ作成方式の場合、n 個の更新データに対し、m 回の冗長データの更新を行なうのに対し、リードモディファイライト方式の場合には1 回の更新に対しm 回の冗長データの更新が発生するからである。このため、できるだけ全ストライプライト方式の冗長データ作成を行なう方が、サブシステム全体のスループットを向上させることに繋がる。第1 および第2 のディスクサブシステムがRAID-5 の記憶方式である場合、第2 のディスクサブシステムへのWRITE/O を起動する第1 のディスク制御装置においては、自サブシステムの冗長データの作成効率を向上させるばかりでなく、第2 のディスク制御装置が効率よく冗長データを作成可能なように二重化のWRITE/O を発行することが、第2 のディスク制御装置のスループットを向上させ、遠隔二重化システム全体のスループット向上に繋がる。

【0017】本発明の目的は、稼働状況に応じてデータ複写の実行方法および契機を制御することにより、複数の記憶サブシステム間でのデータ多重化のためのデータ複写に伴う負荷の増大を抑制して各記憶サブシステムの処理性能を向上させることが可能なデータ多重化記憶サブシステムを提供することにある。

【0018】本発明の他の目的は、稼働状況に応じてデータ複写の実行方法および契機を制御することにより、複数の記憶サブシステム間に設けられたデータ多重化のためのデータ転送経路の負荷の増大を抑制してデータ転送経路の使用効率を向上させることが可能なデータ多重化記憶サブシステムを提供することにある。

【0019】本発明の他の目的は、多重化未完のデータ量の増大を抑止しつつ、複数の記憶サブシステム間でのデータ多重化のためのデータ複写の実行契機の最適化による性能向上を実現することが可能なデータ多重化記憶サブシステムを提供することにある。

【0020】本発明の他の目的は、RAID 等の冗長記憶構成の複数の記憶サブシステム間でのデータ多重化のためのデータ複写に伴う負荷の増大を抑制して各記憶サブシステムの処理性能を向上させることが可能なデータ多重化記憶サブシステムを提供することにある。

【0021】

【課題を解決するための手段】本発明は、第1 のデータ転送経路を介して上位装置と接続される第1 の記憶サブシステムと、少なくとも一つの第2 の記憶サブシステムとを第2 のデータ転送経路にて接続し、第1 の記憶サブシステムが上位装置から受領した書き込みデータを第2

のデータ転送経路を介して第2 の記憶サブシステムに複写することによりデータ多重化を行うシステムにおいて、たとえば第1 の記憶サブシステム内の記憶制御装置に、配下の記憶装置におけるデータの所望の管理単位（たとえばトラック）、もしくは複数の管理単位（たとえばシリンダ等）、またはボリューム単位、ファイル単位毎に一定期間内のデータ更新回数等を記憶するデータ更新回数記憶テーブルを制御情報記憶手段として持つ。

【0022】このテーブルは、たとえば、n 世代前の記録までを保持できるようにn 面のテーブル面を持つ。上位装置からのデータ更新時には、第1 の記憶サブシステムの記憶制御装置の制御プログラム（制御論理）は、最新のデータ更新回数記憶テーブルの当該領域をカウントアップする。このカウントされた値は、当該領域内の第2 の記憶サブシステムへの反映すべきデータの溜り具合を示す指標となる。また、一定周期毎にデータ更新回数記憶テーブルのデータをバックアップ化するとともに最新のデータ更新回数記憶テーブルをクリアする。また別の一定周期毎に、第2 の記憶サブシステムへの未反映データを検索し、多重化のためのWRITE/O 発行契機に、このデータ更新回数記憶テーブルを参照し、n 世代前からの更新回数の変化を調べ、更新回数の減少傾向にあるデータ領域を上位装置からのアクセスが終了しつつあるデータ領域と判断して、増加傾向にあるデータより優先的にサブシステム間のデータ複写をスケジュールするように制御する。一定周期毎の世代管理を行なうことによって、第2 の記憶サブシステムへの未反映データの溜り具合の変化を捉えることが可能となる。

【0023】さらに、第1 の記憶サブシステムの記憶制御装置の制御プログラムは、たとえば、複数の記憶サブシステム間の全データの初期コピー処理による多重化の実行時に、当該多重化の対象範囲（たとえばボリューム）のデータ更新回数記憶テーブルを参照する。第1 の記憶サブシステムの配下のマスタボリュームからのデータの呼び出しおよび第2 の記憶サブシステムへのWRITE/O の発行に際しては、このデータ更新回数記憶テーブルの値と世代間の差による増加・減少傾向から、以降に多重化の対象となる領域の今後の上位装置からの更新を予測し、まだ引き続き更新が継続するようであれば、スケジュールを遅らせる、等の最適化を行う。この様な、データ更新回数の記憶手段とデータ更新の継続性の推測手段によって、初期コピー処理の効率向上を図る。

【0024】一方、各記憶サブシステムの記憶装置が単位データ群と、このデータ群から生成される冗長データを異なる記憶媒体に分散して格納する冗長記憶構成を備えている場合、前記データ更新回数記憶テーブルによる更新履歴から、冗長データを作成するストライプ列に対する更新の継続性を推測する手段を設ける。当該ストラ

イブ列への上位装置からの更新が継続するようであれば、多重化のための第2の記憶サブシステムへのWRT_I/Oの発行スケジュールを遅らせ、全ストライプライト可能な更新データがそろってから、または、更新されていないストライプ列内データをも併せ、ストライプ列全体のデータを纏めて転送する。また、当該ストライプ列への更新の継続が見られない場合には、更新部分のみを転送する。この様な手段によって、第2の記憶サブシステムにおいて、第1の記憶サブシステムから到来する複写データの格納時における全ストライプライトとリ

ードモディファイライトが効率良く制御可能なようにする。

【0025】さらに、各記憶サブシステムの記憶装置が冗長記憶構成を備えている場合、多重化のためのWRT_I/Oの完遂に掛かるデータ転送時間を含むI/O処理時間を観測・記憶する手段を設けることもできる。また、全ストライプ方式による冗長データの作成オーバーヘッドとリードモディファイライト方式による冗長データの作成オーバーヘッドを観測・記憶する手段を設ける。この観測されたI/O処理時間からストライプ列全体のデータ転送に掛かる時間を推測し、リードモディファイライト方式と全ストライプ方式との処理の時間差を比較し、よりオーバーヘッドの少ない方を選択する。

【0026】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照しながら詳細に説明する。

【0027】（実施の形態1）図1は、本発明の一実施の形態であるデータ多重化記憶サブシステムの構成の一例を示す概念図、図2は、本実施の形態のデータ多重化記憶サブシステムにて用いられる制御情報の一例を示す概念図、図3および図4は、本実施の形態のデータ多重化記憶サブシステムの作用の一例を示すフローチャートである。

【0028】まず、図1を用いて本実施の形態のデータ多重化記憶サブシステムの構成例を説明する。中央処理装置101から1ないしはn本のデータ転送パス102で接続されたディスクサブシステムをマスタディスクサブシステム104とし、二重化データを格納するバックアップ側のディスクサブシステムをリモートディスクサブシステム105とする。リモートディスクサブシステム105に接続するバックアップ用CPUをセカンダリCPUとする。それぞれのディスクサブシステムは、ディスク制御装置103と、配下の1ないしn個のディスク記憶装置116とから構成される。このn個のディスク記憶装置116への記憶方式は前述のRAID-5であっても良い。ディスク制御装置103は、1ないしはn個の対CPU制御用プロセッサ107と、1ないしはn個の対リモート制御用プロセッサ113、1ないしはn個の対ディスク制御用プロセッサ114を持つマルチプロセッサにより構成され、それぞれ、対CPUデータ

転送ポート106、対リモートデータ転送ポート112、対ディスクデータ転送ポート115を介して外部とのデータ転送を制御する。この対CPU制御用プロセッサ107と対リモート制御用プロセッサ113は同一のプロセッサでタイムシェアリング的に制御を切り替えるものであっても良い。また、各プロセッサから共通アクセス可能なキャッシュメモリ111と、各プロセッサの共通制御情報を格納する共通メモリ108を持ち、各プロセッサからは共通バス110にてアクセスされる。

【0029】本実施の形態の場合には、この共通メモリ108に、データ更新回数記憶テーブル109を持つ。この様なマスタディスクサブシステム104の構成と同一の構成を持つディスクサブシステムを、たとえば遠隔地に設置してリモートディスクサブシステム105とし、各々のディスク制御装置103の間を1ないしはn本のマスタリモート間データ転送パス117にて接続する。このマスタリモート間データ転送パス117としては、たとえば専用通信回線や公衆通信回線等の任意の情報通信媒体や情報ネットワークを用いることができる。

【0030】次に、中央処理装置101（プライマリCPU）からのマスタディスクサブシステム104へのデータ更新が発生したときの動作例について示す。図2は、データ更新回数記憶テーブル109の構成例である。データ更新回数記憶テーブル109は、二重化対象データの管理単位毎にエントリを持ち（本実施例ではシリンド（CYL）とする）、二重化対象の全領域分のデータを保持する。更に、複数世代にわたってデータ更新回数を管理する場合には、このデータ更新回数記憶テーブル109をn世代分（n面；本実施例では3面、A面～C面（201～203））作成・保持する。最新世代からn世代前までのデータを管理するために、世代管理ポインタテーブル204を持つ。世代管理ポインタテーブル204は、最新世代更新回数記憶テーブルポインタ204a、一世代前更新回数記憶テーブルポインタ204b、二世代前更新回数記憶テーブルポインタ204cからなる。

【0031】次に、上位の中央処理装置101すなわちプライマリCPUからマスタディスクサブシステム104に対するWRT_I/Oを受領したときの対CPU制御用プロセッサ107の制御プログラムの制御（第1の制御動作）について、図3のフローチャートを用いて説明する。ステップ301で上位の中央処理装置101からのWRTコマンドを受領すると、ステップ302でキャッシュメモリ111上のWRTデータ格納領域を確保し、ステップ303で対CPUデータ転送ポート106にCPU-キャッシュメモリ間のデータ転送を起動し、ステップ304でハードウェアのデータ転送完了を待つ。データ転送が完了するとステップ305で世代管理ポインタテーブル204から最新世代更新回数記憶テー

ブルポインタ204aに対応した最新世代の更新回数記憶テーブルA面(201)を得て、ステップ306で当該更新部分に対応する最新のデータ更新回数記憶テーブル109のエントリをカウントアップし、ステップ307で上位の中央処理装置101へWRT_I/Oの終了報告を行なう。当該マスタディスクサブシステム104配下のディスク記憶装置116への実際の書き込みは、本制御を行なっている対CPU制御用プロセッサ107とは異なる対ディスク制御用プロセッサ114の制御によって非同期に書き込まれるものであっても良い。この様に、上位の中央処理装置101からのWRTコマンド受領時に、最新世代のデータ更新回数記憶テーブル109の対応する領域をカウントアップすることによって、データ更新回数の履歴を記憶する。

【0032】次に、対リモート制御用プロセッサ113のデータ二重化のためのWRT_I/O発行処理に関する制御プログラムの制御の一例について、図4および図5を用いて説明する。図4は主にデータ更新回数記憶テーブル109の制御に関する処理フローの例である。対リモート制御用プロセッサ113の制御プログラムは、ダイナミックにループしながら特定周期毎にデータ更新回数記憶テーブル109の管理とリモートディスクサブシステム105へのWRT_I/OのスケジュールおよびWRT_I/O実行処理を行なう。ここでステップ401の周期Aは、データ更新回数記憶テーブル109の制御を周期的に行なう処理の起動周期であり、ステップ402の周期Bは二重化WRT_I/Oスケジュールおよび実行処理の起動周期である。周期Aと周期Bは、A>Bの関係の適当な周期とする。ステップ401で周期Aの経過を検知すると、データ更新回数記憶テーブル109の管理の処理を行なう。すなわち、ステップ403で、世代管理ポインタテーブル204の二世世代前のデータ更新回数記憶テーブル109を指すポインタテーブル(二世世代前更新回数記憶テーブルポインタ204c)の値を一時的にワークエリアに退避し、ステップ404で一世代前のデータ更新回数記憶テーブル109を指すポインタ値(一世代前更新回数記憶テーブルポインタ204b)を二世世代前のデータ更新回数記憶テーブル109を指すポインタ格納領域(二世世代前更新回数記憶テーブルポインタ204c)に複写する。ステップ405で最新のデータ更新回数記憶テーブル109を指すポインタ値(最新世代更新回数記憶テーブルポインタ204a)を一世代前の更新回数記憶テーブル109を指すポインタ(一世代前更新回数記憶テーブルポインタ204b)に複写する。ステップ406でワークエリアに退避してあった元の二世世代前の更新回数記憶テーブルへのポインタ値を最新世代更新回数記憶テーブルポインタ204aの格納領域に複写し、ステップ407で最新世代のデータ更新回数記憶テーブル109となったテーブル面をクリアし最新化を図る。この様にして、特定周期でデータ

更新回数記憶テーブル109の複数面を入れ替え世代管理を実現する。ステップ402で周期Bの経過を検知すると二重化WRT_I/Oスケジュールおよび実行処理を行なう(ステップ410)。

【0033】この二重化WRT_I/Oスケジュールおよび実行処理の一例を示すフローチャートを図5に示す。ステップ501からステップ505がマスタディスクサブシステム104上に溜まっているリモートディスクサブシステム105への未反映データの検索処理である。この処理において、本実施の形態にて例示されるデータ更新回数記憶テーブル109等の手段によって、更新回数の時系列的な変化を捉え、WRT_I/O実行の対象とするか否かの判断を行なう。具体的には、ステップ501でキャッシュメモリ111上のリモートディスクサブシステム105への未反映データを検索する。この検索は、たとえばハッシュを用いたキャッシュメモリ111上のデータ管理方法によるものであっても良いし、LRUアルゴリズムによるものであっても良い。ステップ502で検索された未反映データの領域に対応する更新回数を、各世代毎のデータ更新回数記憶テーブル109から読み出す。ステップ503で、二世世代前の更新回数から一世代、最新世代と比較し、増減傾向を調べる。比較結果から当該領域への更新が増加傾向にあると判断できる場合には、この時点での当該データのスケジュールを見送り別の未反映データの検索処理を行なう。また、全未反映データの検索が終了した場合には、もとの処理に戻る(ステップ504~505)。ステップ504で当該未反映データの範囲に対する更新が増加傾向に無い場合は、当該領域を二重化WRTの対象とし、リモートディスクサブシステム105へWRT_I/Oを発行する(ステップ506~510)。

【0034】次に、この二重化WRT_I/O実行処理について一例を記す(ステップ506~510)。

【0035】まず、当該未反映データに対応するリモートディスクサブシステム105のデバイスに対しWRTコマンドを発行する(ステップ506)。

【0036】ステップ507で当該WRTコマンドに伴うデータ転送の完了待ちを行い、転送完了後、ステップ508でリモートディスクサブシステム105からのI/O完了報告待ちを行う。そしてステップ509にてエラー判定を行い、このI/O完了報告が異常終了であった場合には、ステップ510のエラー処理を行い、正常終了であった場合には元の周期監視処理に戻る。

【0037】このように、ランダム性に富んだ上位の中央処理装置101からのディスクデータの更新に際しては、そのままでは正確な予測は不可能であるが、本実施の形態のデータ多重化記憶サブシステムによれば、たとえば、多重化のためのWRT_I/O発行契機に、データ更新回数記憶テーブル109を参照し、n世代前からの更新回数の変化を調べ、更新回数の減少傾向にあるデ

10

20

30

40

50

ータ領域を上位の中央処理装置101からのアクセスが終了しつつあるデータ領域と判断して、増加傾向にあるデータより優先的にマスターリモート間データ転送パス117を経由したサブシステム間のデータ複写をスケジュールするように制御することで、各データ領域における更新状況に応じた必要最小限のデータ複写回数にてデータの二重化を効率よく達成することが可能となる。また、一定周期毎の世代管理を行なうことによって、リモートディスクサブシステム105への未反映データのキャッシュメモリ111等における溜り具合の変化を捉えることが可能となり、データ複写の遅延に起因するキャッシュメモリ111等の利用効率に低下を回避して、システムのキャッシュメモリ111等の資源の可用性の向上を実現できる。

【0038】さらに、データ複写のためのマスターリモート間データ転送パス117を経由した無駄なデータ転送が減り、マスターリモート間データ転送パス117を構成する情報通信媒体や情報ネットワーク等の負荷(トラヒック)の増大を防止して、情報通信媒体や情報ネットワーク等の効率的な利用によるデータ多重化が可能になる。

【0039】(実施の形態2)次に、図1に例示される構成の本発明のデータ多重化記憶サブシステムにおいて、マスタディスクサブシステム104とリモートディスクサブシステム105との間における初期コピー操作等における効率的なデータ複写の実現方法(第2の制御動作)の一例を、図6のフローチャートにて説明する。

【0040】たとえば、図1に示すマスタディスクサブシステム104上のボリュームデータを新たに二重化ペアとして設定する場合、マスタディスクサブシステム104のディスク制御装置103は、自配下の当該ボリュームデータをキャッシュメモリ111上にステージングし、そのデータをWRT_I/Oによってリモートディスクサブシステム105へ書き込む。この動作を当該ボリュームの全トラック(TRK)範囲に渡って繰り返すことによって初期コピー(初期のボリュームデータ多重化)を行なう。自配下のボリュームからのステージング処理は図1に示す対ディスク制御用プロセッサ114が制御し、リモートディスクサブシステム105へのWRT_I/Oの発行は対リモート制御用プロセッサ113が制御する。

【0041】図6に例示されるフローチャートは、初期のボリュームデータ多重化にて複写対象領域に対する上位の中央処理装置101からのデータ更新要求の有無や頻度等に応じてデータ複写の順序を動的に変更する機能を持った対リモート制御用プロセッサ113の初期コピー処理例である。ここで、本実施の形態に示すデータ更新回数記憶テーブル109の世代管理については、先の図2および図3に例示した場合と同様の制御を行う。対リモート制御用プロセッサ113は、初期コピー処理に

おいては、初期コピー対象のデバイスを選択し(ステップ601)、対象のデバイスの次コピー対象領域の各世代毎のデータ更新回数記憶テーブル109の値を読み出す(ステップ602)。ここで1回のリモートディスクサブシステム105へのコピーの単位は複数のデータ単位(トラック)であるシリンダ単位であっても良いし、またトラック単位であっても良い。ステップ603で各世代毎の更新回数を比較し、もし次コピー対象領域に対する上位からの更新が増加傾向にある場合には、当該ボリュームの今回のコピー処理を見合わせ、もし初期コピー処理が複数個のボリュームで同時になされている場合には、別の初期コピー対象のデバイスの初期コピー処理のスケジュールを行なう(ステップ604およびステップ613)。ステップ604で次コピー対象領域への更新が増加傾向にない場合は、当該領域を次コピー対象領域と定めコピー対象範囲内のトラックがキャッシュメモリ111上に存在する(Cache Hit)か存在しないか(Cache Miss)を検索する(ステップ605)。ステップ606でCache Missであるトラックのステージング処理要求を対ディスク制御用プロセッサ114に発行し、ステップ607でそのステージング処理完了待ちを行なう。コピー対象領域のステージング処理が完了し、全トラックがキャッシュメモリ111上にステージングされている状態で二重化のためのWRT_I/O発行をリモートディスクサブシステム105に行なう(ステップ608から612)。この処理は、先に図5に例示した実施例の処理と同一である。ステップ611にて二重化のためのWRT_I/O処理が正常に終了すると、ステップ613で別の初期コピー対象デバイスを選択し、これまでと同様の制御を繰り返し、初期コピー処理を完成させる。

【0042】このように、本実施の形態の場合には、たとえばシステム立ち上げ時等の契機にて実行される、マスタディスクサブシステム104の配下のディスク記憶装置116の全データをリモートディスクサブシステム105に複写する初期データ複写等において、データ複写中に複写予定のデータ領域に上位からの更新要求が集中するような場合には、当該更新要求が無くなるまで当該領域の複写を後回しにして、他の更新要求が発生していない安定なデータ領域の複写を先行させる等の制御を行うことで、初期データ複写等の処理を効率よく遂行することが可能になる。

【0043】(実施の形態3)次に、図1に例示される構成のデータ多重化記憶サブシステムにおいて、RAID方式にて、マスタディスクサブシステム104およびリモートディスクサブシステム105にてデータを格納する場合のデータ二重化のためのデータ複写の効率化の一例について説明する。

【0044】図7に本実施の形態におけるRAID方式のデータ格納の例を示す。RAID方式によるデータの

10

20

30

40

50

記憶は、論理ボリュームを $n+1$ 個のデバイスで構成し、データの格納単位(たとえばトラック)毎に、デバイスを分けて格納する。データ(トラック)は n 個の単位でストライプを成し、それに対し一つまたは複数の冗長データを持つ。図7に示す格納例では、トラック番号0から $n-1$ までのトラックを1ストライプとし一つの冗長データ(Parity #0)を持つ。同様にトラック $\#n \times 1$ から $\#n \times 1 + (n-1)$ までのストライプに対しParity #1を持つ。ここで冗長データの配置は常に固定のデバイスに配置する方式であっても良いし、図7の例に示すように順次に格納デバイスを変えて格納する方式であっても良い。RAID方式におけるデータの更新は、前述の様に全ストライプ列で冗長データを作成する全ストライプライト方式と、更新データとその旧データおよび旧冗長データから新冗長データを作成するリードモディファイライト方式とがある。冗長データを構成するストライプ列中の更新データが複数個に渡る場合には、全ストライプライト方式で冗長データを作成する方がそのオーバーヘッドは削減される。

【0045】まず、特定のストライプ内の各データ単位(この場合はトラック)の各々に対する更新要求の発生状況に応じて、複写先のリモートディスクサブシステム105に対して、更新されたトラックのデータのみを転送してリードモディファイライト方式でのデータ格納を促す(第2の複写操作)か、更新トラックを含む全ストライプデータの転送によって、全ストライプライト方式でのデータ格納を促す(第1の複写操作)かを動的に切り換える場合(第3の制御動作)について説明する。

【0046】すなわち、二重化のためのWRT_I/O発行対象のデータのストライプ範囲の更新回数の世代毎の更新履歴から増加傾向を判断し、増加傾向にある場合には当該ストライプ列の更新が継続するとの判断から、ストライプ列での更新を促すようにストライプ列範囲全般に渡って更新がなされるまで、WRT_I/Oの発行を遅らせる。減少傾向にある場合、または更新の継続性が見られない場合には、当該データをスケジュールした時点で二重化のためのWRT_I/Oの発行を行なう。さらに、この場合更新された部分のみを二重化のためのWRT_I/Oで転送してもよいし、更新はされていない同一ストライプ列の他のデータを併せてWRT_I/Oでリモートディスクサブシステム105に書き込むことによって、リモートディスクサブシステム105にて全ストライプライトが促進されるように制御することも可能である。

【0047】上述の例では、ストライプ内の各トラックに対する更新要求の有無にてストライプ内の全トラックをまとめて転送するか、更新のあったトラックのみを転送するかを切り換えていたが、さらに、この切り換えの判定条件として、マスタディスクサブシステム104からリモートディスクサブシステム105へのデータ転送

時間を含めたデータ複写の全所要時間の大小に応じて、リードモディファイライト方式か、全ストライプライト方式かを選択させる場合(第4の制御動作)の一例について、図8および図9を参照して以下に説明する。データ転送の遅延時間は、 5 ns/m なので、複数の記憶サブシステムがたとえば数百キロも離れた遠隔地に設置された場合、両者間におけるデータ転送所要時間は各サブシステム内の処理時間に比較して無視できないほど大きくなり、このデータ転送所要時間を加味して、データ転送方法を切り換えることは、データ複写処理の効率改善において大きな意味を持つ。

【0048】すなわち、この判断基準に二重化のためのWRT_I/Oに要するトラック単位の処理オーバーヘッドを観測する手段として、図1に示す共通メモリ108に、図8に例示される構成の平均WRT_I/O処理時間格納テーブル801と平均値観測回数カウンタ802(平均値観測回数: N)を設ける。

【0049】図9にトラック単位の処理オーバーヘッドの観測のフローチャートの一例を示す。平均WRT_I/O処理時間(t_m)を観測する二重化のためのWRT_I/O発行処理の基本的な流れは、図5に示すものと同様である。この図5に例示された処理の流れに加えて図9の例では、更に、ステップ911でデータ転送開始時刻を記憶し、ステップ914でI/O完了報告が正常になされたか否かを判定した後、ステップ916で現時刻とデータ転送開始時刻との差からWRT_I/O処理時間(Δt)を算出する。ステップ917で、これまでの平均WRT_I/O処理時間(t_m)と今回計測されたWRT_I/O処理時間(Δt)とから最新の平均WRT_I/O処理時間(t_m)を算出する。また、ステップ918で算出した最新の平均WRT_I/O処理時間(t_m)を平均WRT_I/O処理時間格納テーブル801に格納し、ステップ919で平均値観測回数カウンタ802(N)をインクリメントする。尚、ステップ917の算出(平均WRT_I/O処理時間(t_m)の更新)方法は、たとえば下記の(式1)に例示される通りである。

【0050】

【数1】

$$t_m \leftarrow \frac{t_m \times N + \Delta t}{N + 1} \quad (\text{式1})$$

【0051】上記のようにして平均WRT_I/O処理時間(t_m)を観測し、このWRT_I/O処理オーバーヘッドを元に全ストライプ列のデータを併せて、リモートディスクサブシステム105に更新を行なった方がトータルの処理時間が削減されるか否かを判断する。この判断ステップは、たとえば、図9におけるステップ910の直前に、平均WRT_I/O処理時間(t_m)の推移に応じて、判定ルーチンを実行することで実現できる。たとえば、リードモディファイライト方式、および

全ストライプライト方式の各々において観測された平均WR T _I /O処理時間(t m)の大小を比較し、t mがより小さい方式を選択してデータ複写を実行する、等の方法が考えられる。

【0052】このように、マスターリモート間データ転送パス117におけるデータ転送時間を含めたデータ複写処理の全所要時間を観測することにより、マスタディスクサブシステム104およびリモートディスクサブシステム105にてRAID方式でデータを格納する場合において、データ複写先のリモートディスクサブシステム105でリードモディファイライト方式および全ストライプライト方式のいずれの方式でデータ格納動作を行わせるかをよりきめ細かく制御でき、データ多重化のためのデータ複写の最適化および効率化を実現することが可能になる。

【0053】以上本発明者によってなされた発明を実施の形態に基づき具体的に説明したが、本発明は前記実施の形態に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。

【0054】

【発明の効果】本発明のデータ多重化記憶サブシステムによれば、稼働状況に応じてデータ複写の実行方法および契機を制御することにより、複数の記憶サブシステム間でのデータ多重化のためのデータ複写に伴う負荷の増大を抑制して各記憶サブシステムの処理性能を向上させることができる、という効果が得られる。

【0055】また、本発明の、データ多重化記憶サブシステムによれば稼働状況に応じてデータ複写の実行方法および契機を制御することにより、複数の記憶サブシステム間に設けられたデータ多重化のためのデータ転送経路の負荷の増大を抑制してデータ転送経路の使用効率を向上させることができる、という効果が得られる。

【0056】また、本発明のデータ多重化記憶サブシステムによれば、多重化未完のデータ量の増大を抑止しつつ、複数の記憶サブシステム間でのデータ多重化のためのデータ複写の実行契機の最適化による性能向上を実現することができる、という効果が得られる。

【0057】本発明のデータ多重化記憶サブシステムによれば、RAID等の冗長記憶構成の複数の記憶サブシステム間でのデータ多重化のためのデータ複写に伴う負荷の増大を抑制して各記憶サブシステムの処理性能を向上させることができる、という効果が得られる。

【図面の簡単な説明】

【図1】本発明の一実施の形態であるデータ多重化記憶サブシステムの構成の一例を示す概念図である。

【図2】本発明の一実施の形態であるデータ多重化記憶サブシステムにて用いられる制御情報の一例を示す概念図である。

【図3】本発明の一実施の形態であるデータ多重化記憶サブシステムの作用の一例を示すフローチャートである。

【図4】本発明の一実施の形態であるデータ多重化記憶サブシステムの作用の一例を示すフローチャートである。

【図5】本発明の一実施の形態であるデータ多重化記憶サブシステムの作用の一例を示すフローチャートである。

【図6】本発明の一実施の形態であるデータ多重化記憶サブシステムの作用の一例を示すフローチャートである。

【図7】本発明の一実施の形態であるデータ多重化記憶サブシステムにおけるRAID方式のデータ格納方法の一例を示す概念図である。

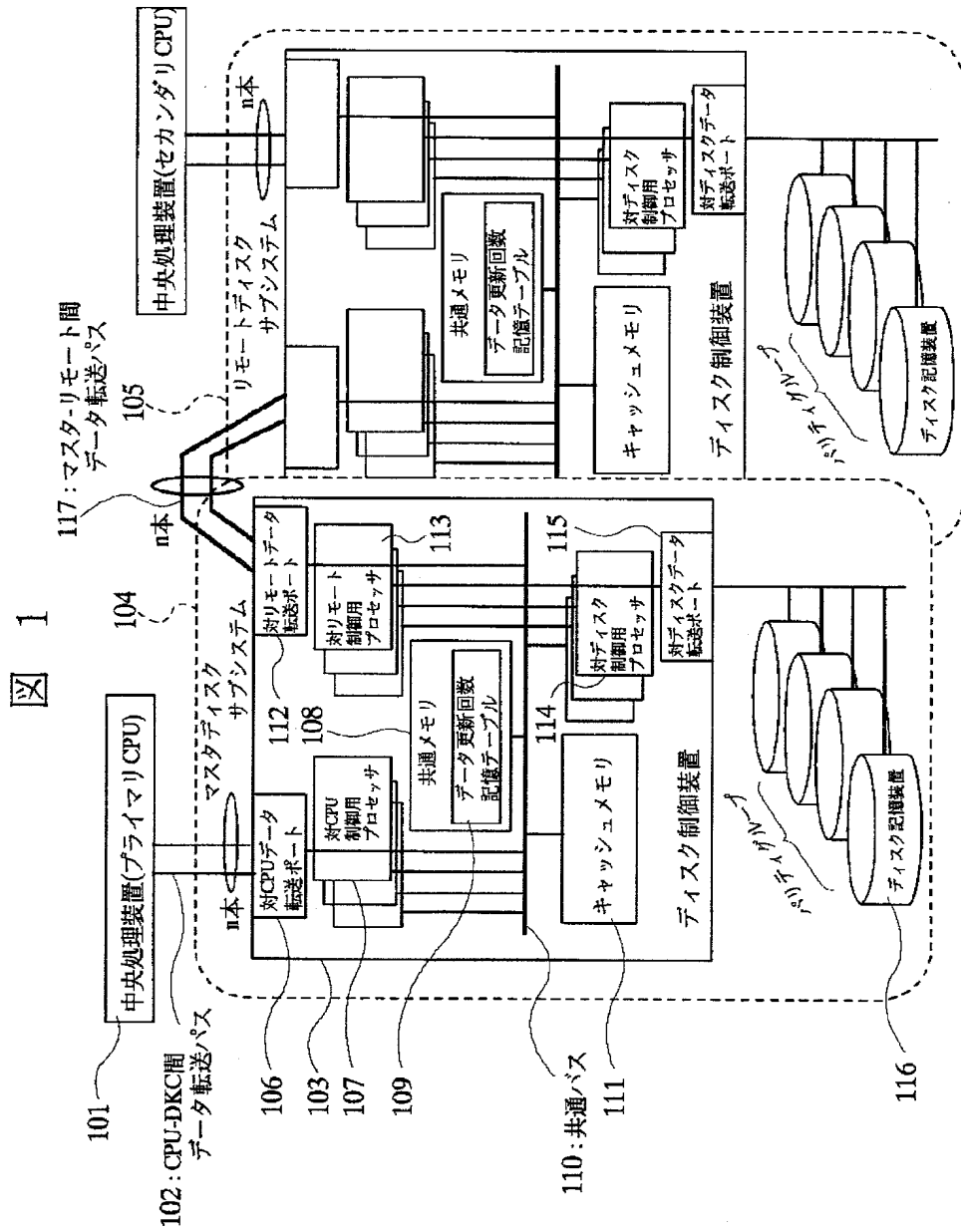
【図8】本発明の一実施の形態であるデータ多重化記憶サブシステムにて用いられる制御情報の一例を示す概念図である。

【図9】本発明の一実施の形態であるデータ多重化記憶サブシステムの作用の一例を示すフローチャートである。

【符号の説明】

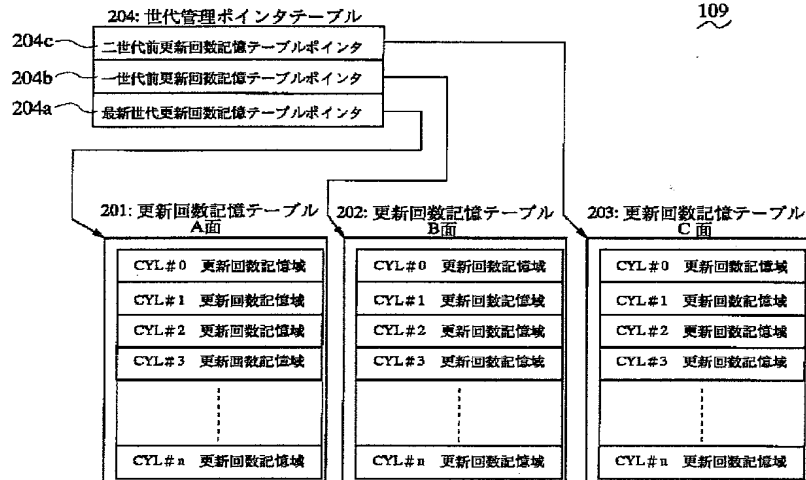
101…中央処理装置(上位装置)、102…データ転送バス(第1のデータ転送経路)、103…ディスク制御装置、104…マスタディスクサブシステム(第1の記憶サブシステム)、105…リモートディスクサブシステム(第2の記憶サブシステム)、106…対CPUデータ転送ポート、107…対CPU制御用プロセッサ、108…共通メモリ、109…データ更新回数記憶テーブル(制御情報記憶手段)、110…共通バス、111…キャッシュメモリ、112…対リモートデータ転送ポート、113…対リモート制御用プロセッサ、114…対ディスク制御用プロセッサ、115…対ディスクデータ転送ポート、116…ディスク記憶装置、117…マスターリモート間データ転送バス(第2のデータ転送経路)、204…世代管理ポインタテーブル、204a…最新世代更新回数記憶テーブルポインタ、204b…一世代前更新回数記憶テーブルポインタ、204c…二世代前更新回数記憶テーブルポインタ、801…平均WR T _I /O処理時間格納テーブル、802…平均値観測回数カウンタ。

【 図1 】



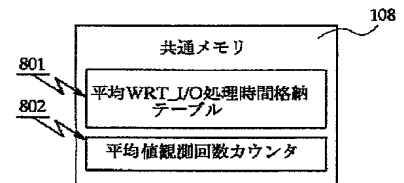
【 図2 】

図 2



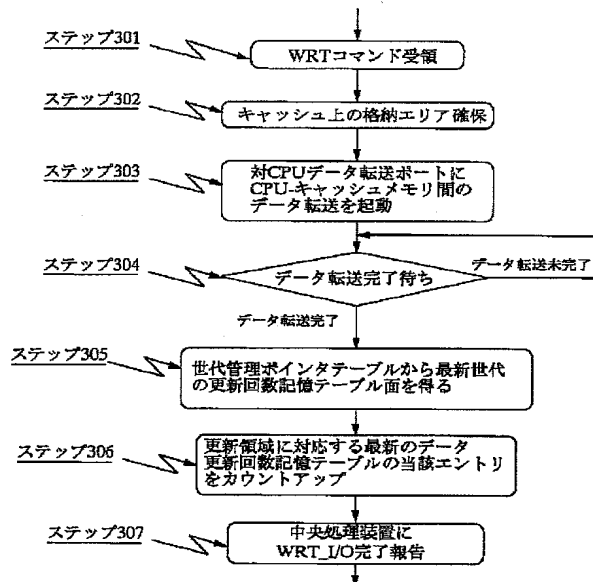
【 図8 】

図 8



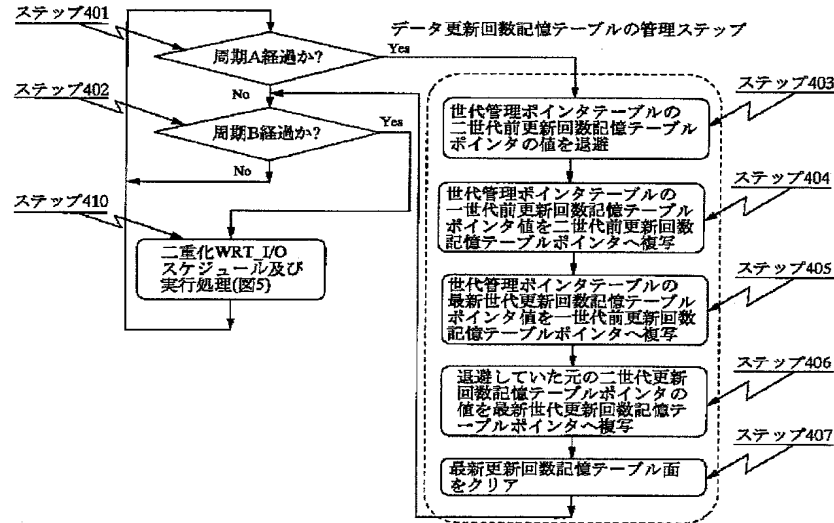
【 図3 】

図 3



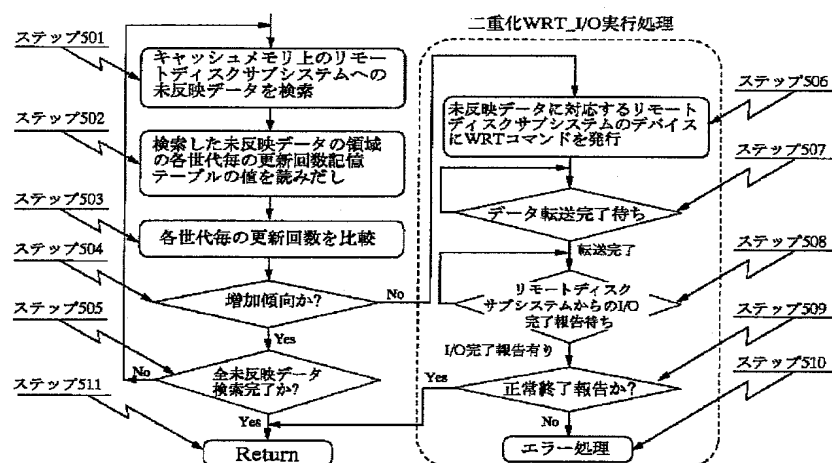
【 図4 】

図 4

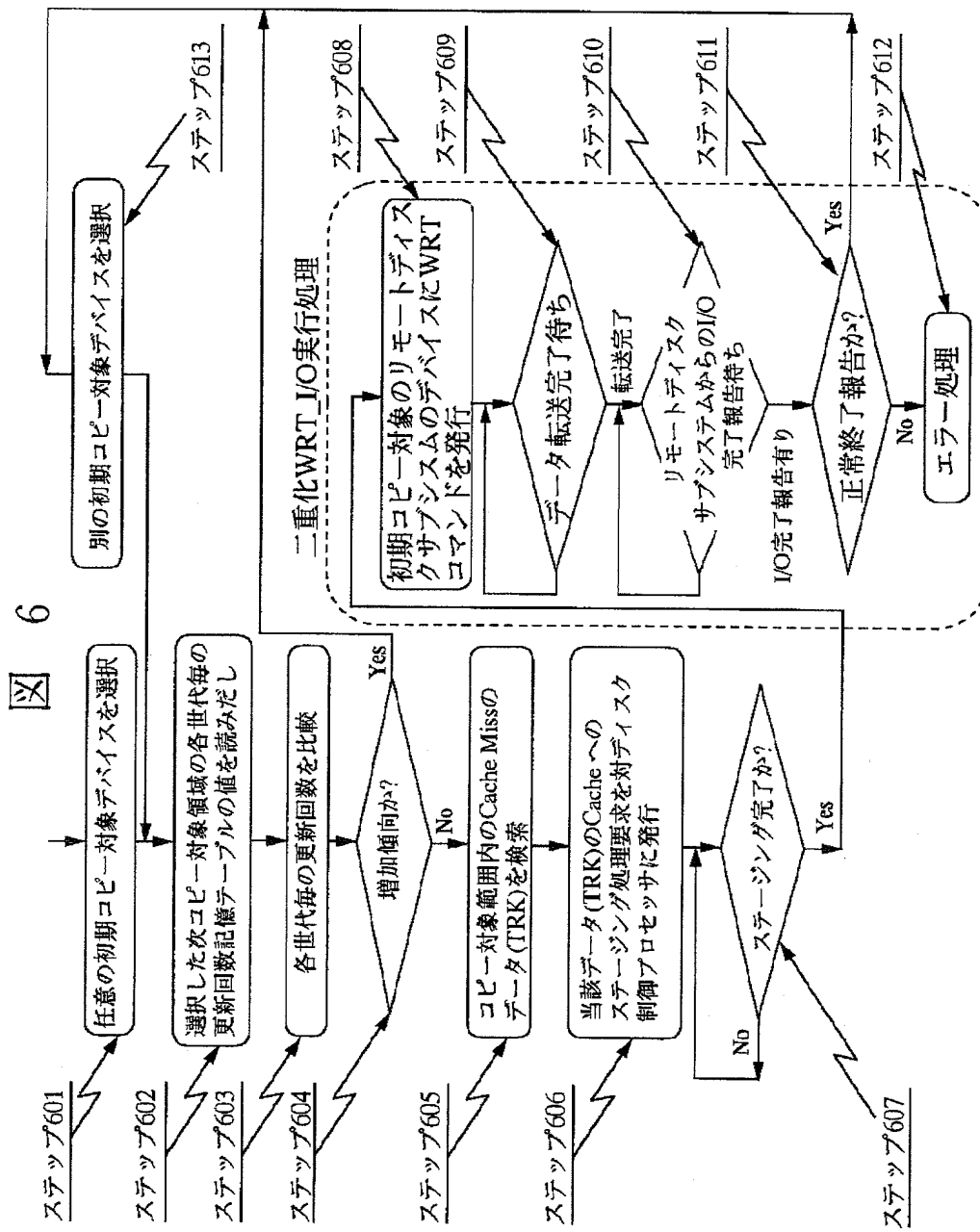


【 図5 】

図 5

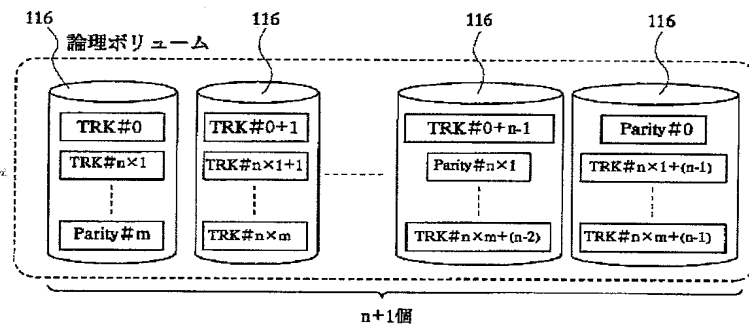


【 図6 】



【 図7 】

図 7



【 図9 】

図 9

